UTILIZING MACHINE LEARNING ALGORITHMS FOR PREDICTING BOOK POPULARITY SCORES USING TRANSACTION HISTORY AND BOOK CHARACTERISTICS FOR PUBLIC LIBRARIES

WORD COUNT: 8,516

CAS JULICHER

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2064352

COMMITTEE

dr. Görkem Saygili
dr. Boris Čule

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

June 24, 2022

UTILIZING MACHINE LEARNING ALGORITHMS FOR PREDICTING BOOK POPULARITY SCORES USING TRANSACTION HISTORY AND BOOK CHARACTERISTICS FOR PUBLIC LIBRARIES

WORD COUNT: 8,516

CAS JULICHER

Abstract

This thesis investigates the possibility to produce library-specific book popularity scores. It will do so by using book characteristics, transaction and inventory records of 5 libraries in the Netherlands. Additionally, feature importances are evaluated per library. This proposed score can manipulate a generic score that is currently implemented in the library system, as this score is not library-specific and not informative enough. The algorithms: XGBoost, Support Vector Machine and a Logistic Regression classifier will be used. The data set used does not include the necessary data such as user ratings to accomplish results that similar studies achieved. This thesis is therefore unique by creating its target label, using transaction and inventory records. Due to the characteristics of the target label class imbalance is introduced. Therefore, SMOTE oversampled the training set (SMOTE-experiment) and the performance is compared to the original training set (Baseline-experiment). The performance varies across libraries, but XGBoost is the best overall performing model in both experiments. Additionally, the class-specific performance made evident that the artificial instances, in the SMOTE-experiment, did not have the desired effect within the training of the models. They did not represent the properties of the class well, this is supported by the Baseline-experiment. Before this method can be implemented, challenges regarding the target creation should be solved. As the target variable should be balanced, using other over-or undersampling techniques, changing the construction of the target label or including user ratings should be considered. Finally, numeric variables score higher than categorical variables in feature importance. This is not the best method and the field of explainable AI should be included.

CONTENTS

1 DATA SOURCE AND ETHICS

Work on this thesis did not involve collecting data from human participants or animals. It does not include any personal data and is not traceable to a person in any way. The data originates from NBD Biblion, a client of Enjins, which is also the external partner for this thesis. NBD Biblion gave permission to use the data for this thesis. However, they will retain ownership of this data. The author of this thesis acknowledged that there is no legal claim to this data. The code and data used in this thesis are not publicly available but are available for the professors from Tilburg University that are supervising this thesis and they can access the code via this GitHub repository[1]. It can however be made available upon request via this email address[2] if both Enjins and NBD Biblion approve this.

[1] https://github.com/Julicher/Thesis-DSS.git
[2] cas.julicher@gmail.com

2 INTRODUCTION

This thesis, in collaboration with Enjins and NBD Biblion, will focus on the prediction of book popularity scores within public libraries in the Netherlands. These popularity scores can be used to predict how well a book will perform in a public library compared to that same book in other libraries. This will be accomplished by using book characteristics, transaction and inventory records of public libraries. The transaction and inventory records represent the behaviour of library customers, also referred to as lending behaviour, which is used as the target variable. The algorithms XGBoost (XGB), Support Vector Machine (SVM) and Logistic Regression (LR) will be utilized based on various literature, as emphasized in Section 3.

The current score that indicates the popularity of a book is not library-specific. It is a generic popularity score that all public libraries use ranging from one to ten, with one indicating a low popularity score and ten indicating a high popularity score. This means that it is the same for every library. Although, every library has a different customer demographic and focus (NBD Biblion, 2022). This generic score is based upon sales estimations, reviews by critics, actuality and the experience of critics. The generic score thus holds remarkable value for users of the said score. Nevertheless, the generalizability of the score is not at level for every individual library because of the extensive scope the metric tries to cover.

NBD Biblion, the distributor of books for almost all public libraries in the Netherlands, is introducing a new feature namely data-driven collections. In their web shop, they present a selection of books based on the personal preferences, budgets and buying history. The generic popularity score is currently implemented in this service, as seen in Appendix A. This thesis proposes a method to manipulate this generic score, by applying machine learning to library data. This will create a data-driven score that is library-specific and as informative as possible by combining the generic score and the data-driven based score. This thesis will not manipulate the generic score as an end goal but proposes a method for achieving this for future applications within data-driven collections at NBD Biblion.

This thesis will contribute on a scientific level as this is innovative due to the novel characteristics investigated here, as the problem has never been addressed in physical public libraries and the data set has not been used before as it is not publicly available. Thus, the possibility that lending behaviour in combination with book characteristics can be used to produce library-specific book popularity scores will be explored. Additionally, it

will be investigated if there are noteworthy differences between libraries in what characteristics they value most, this is analysed on the best performing model. This thesis contributes on a societal level as it tries to find a relationship between lending behaviour and book characteristics. This can be used to create an optimal inventory and buy-in procedure for physical public libraries and thus facilitating their customers better. It also contributes to the implementation of data science methods in physical stores, as in general most data-driven solutions apply to online stores (Thiebaut, 2019).

To achieve the desired result of creating personalized book popularity scores on a library level, this research is driven by a main research question and supported using two sub-questions.

> *To what extent can book characteristics in combination with lending behaviour explain library-specific book popularity scores?*

The sub-questions can be listed separately, as such:

SQ1 *How accurate can XGBoost predict book popularity scores compared to a Logistic Regression classifier and Support Vector Machine classifier across different libraries?*

SQ2 *How do book characteristics and lending behaviour affect the predictions across libraries?*

The first sub-question will be answered by comparing the performance of three traditional machine learning algorithms. The second sub-question will be answered by retrieving feature importances from the best performing model, these are compared to each library and the findings will be reported. The two sub-questions will assist in answering the main research question by using performance measures and it is decided if the performance is sufficient enough that lending behaviour can be explained using book characteristics.

2.1 *Structure of this thesis*

To achieve the desired result, this thesis will adhere to the following structure. First, the related work is discussed (Section 3), this outlines the landscape in which this research will operate in. It will analyse methods that are used on similar problems, such as online book stores and identify gaps in the literature. The methodology and experimental setup (Section 4) will demonstrate the data, data cleaning, label creation, model selection, tuning of the models and the evaluation metrics. This will be followed by the results (Section 5), which will be analysed and this leads into the discussion (Section 6) and conclusion (Section 7) where the results are discussed and a conclusion will be provided concerning the research question.

2.2 *Main findings*

The method of data-driven popularity scores will not work for every library as the performance is highly correlated with the availability and quality of the data. Due to the construction method of the target variable, class imbalance is introduced, which introduced challenges in resolving and resulted in remarkable performance differences between classes. It should be investigated if this can be resolved with sufficient data or a different construction of the target label. It is also important to look into the possibility to collect user ratings as a library, as then the performance would increase according to literature. Feature importances should be handled with care and different approaches such as including the field of explainable artificial intelligence. This can assist in achieving the desired result. The main contribution of this thesis is that there is an improved understanding of library data, the technical library landscape and challenges that have to be solved before the proposed data-driven score can be implemented within the library system.

3 RELATED WORK

In this section, various scientific papers will be discussed and it will adhere to the following structure. Firstly, the position of a public library in this technological area will be discussed, followed by (theoretical) applications of data science within the public library system. This will include papers that implemented data science projects in libraries. It also examines how online book stores recommend books and estimate popularity. This includes the differences and challenges between online and physical libraries regarding their models choice on similar problems. As this problem has never been addressed before there is no direct comparable material that this thesis can be based upon. However, this section explores comparable (conceptual) works that examine similar problems and models from different standpoints.

3.1 *The position of a library in this technological era*

Data science started to emerge within large companies to create revenue, automate tasks and connect more easily with other parts of the world. Nowadays everyone is involved in data science, even if they are not aware of their participation. It is even seen as a new era; the 4^{th} industrial revolution (Skilton & Hovsepian, 2017). However, the use of data science varies among companies, as they all have different objectives and funding. Public libraries have multiple objectives as governments want them to be in the centre of this digital era, as they play an important role in education, social support of communities and cultural appreciation. Additionally, public libraries should be a free and open space accessibly to everyone (Howard, 2019).

Taking a position in this technical landscape is in practice not easy (Leorke, Wyatt, & McQuire, 2018) as the field is developing rapidly and there is an absence of funding and knowledgeable people (Leorke et al., 2018). Zhan and Widén (2018) strengthen this point by interviewing library personnel to validate if they understand and can identify data careers within a library. The outcome was that library personnel are in agreement that these roles have value but they lack the knowledge to recognise the importance and the content that these roles have. This highlights that in practice the implementation of data science is not as easy as thought, as current employees still have a hard time grasping the contents of these developments and need training in trusting and using data-driven applications (Cao, Liang, & Li, 2018).

Public libraries have a great deal of data, such as transactions, inventory, book characteristics and customer information. This could be of value to play into book trends, maintain an optimal inventory and automate processes across libraries. As stated earlier, libraries have various sources of data and according to Teets and Goldner (2013) libraries are ready to use big data. Big data can be defined using the following three components: variety, velocity and volume (Sagiroglu & Sinanc, 2013). These components are referring to data variety in the sense of structured, unstructured and semi-structured data. Velocity in respect of data gathering, is it streaming data, real-time or is it received in batches. Volume concerns the size of the data. Sagiroglu and Sinanc (2013) state that data is big data when a combination of these components is met. Libraries are dealing with big data as there is enough volume and various varieties and velocities of data (Teets & Goldner, 2013). Teets and Goldner (2013) also explain that libraries should take on a specific function in the data landscape, by sharing data and models across libraries and helping the public library system to improve in the data landscape.

3.2 *Physical and online libraries and artificial intelligence*

Artificial Intelligence (AI), is a part of computer science that researches the possibility to make machines human-like (An, Li, Su, & Wang, 2021). AI can be used to develop a system to make public libraries smarter (Otterlo, 2017). Otterlo (2017) suggested that libraries could integrate AI using the following workflow: Data gathering – Analysis – Intervention – Feedback. Otterlo (2017) compares this workflow to how Amazon recommends its products online. This thesis will focus on the intervention section of the workflow from Otterlo (2017), as it is not feasible to implement the other parts of the workflow in the scope of this thesis. However, it is important to describe the whole workflow as this provides a background to the process of making public libraries data-driven.

At Amazon, the workflow of Otterlo (2017) is automated as customers click on books (Data gathering), this is analysed and compared to other customers that follow the same pattern (Analysis), Amazon then recommends books to customers based on that information (Intervention) and the customer buys the books and perhaps writes a review (Feedback). At the start of Amazon, an item-to-item collaborative filtering algorithm was designed by Linden, Smith, and York (2003), which is a recommender system. A recommender can be seen as a system that assists in the decision-making regarding what users like. Even if there is not enough data on a user, it looks at similar users and recommends items based on that information (Aggarwal, 2016). To this day, with updates to the model, the recommender

by Linden et al. (2003) still outperforms several neural networks in their recommendation strength in some product areas of Amazon and is still state of the art (Hardesty, 2022). This demonstrates the impact recommenders have, particularly when Amazon accounts for 42% of all book sales in the United States (Palumbo, 2019). If public libraries succeed in the integration of a similar workflow that Otterlo (2017) suggested, data-driven collections and recommendations could improve considerably. Recommender systems are an important method in E-commerce to recommend products. However, they cannot be implemented in this thesis as the data used does not include user ratings, user considerations and a large customer base.

In contrast to online book stores, physical libraries have more challenges to face when implementing a data-driven workflow, especially with data gathering. It is hard to physically track customers and analyse which books they consider before lending the final book, also regarding data ethics and privacy of the customer.

3.3 *Data science practices on library data*

Hicks, Cavanagh, and VanScoy (2020) tried to identify the community of a library using a social network analysis and map the interactions a library has with its community. This is a part of data gathering in the workflow of Otterlo (2017), as it is crucial to identify customers. Tsuji et al. (2012) analysed various methods to recommend books using collaborative filtering on loan records, association rule mining and the recommender from Amazon. They found that the Amazon algorithm was the most effective in recommending followed by association rule mining and collaborative filtering. This emphasises the importance of logging user actions and considerations instead of only relying on the loan records.

Lee, Ji, Kim, and Park (2021) investigated the possibility to identify best-seller books using machine learning, based on book characteristics and user ratings for an online library. They used various classification algorithms such as Logistic Regression, Support Vector Machines, Naïve Bayes, and Random Forest. They achieved an average F1 score 0.81 when classifying across five categories. This research by Lee et al. (2021) investigates a comparable problem to the current scope of the thesis, but they used a critical part of data that is not available within this thesis, user ratings, which was their target. Lee et al. (2021) also compared their results to recommendation algorithms that are discussed in the previous paragraph. They found that collaborative filtering systems have limitations, such as data sparsity and cold starts (Çano & Morisio, 2017). Consequently, content-based filtering

cannot use the content of a book, as issues regarding copyright could arise.

Research by Ullah et al. (2021); Çolakoglu and Akkaya (2019) used the combination of XGB, LR and SVM in a classification study and both yielded high performance of the XGB model compared to the SVM and LR classifiers. As Lee et al. (2021) used SVMs and a LR on a similar problem, these models are used to compare the performance with XGB. XGB is the primary model as in literature it yields high performance on relatable data sets, this is further emphasized in Section 4.

3.4 *Gaps in the literature*

In general, there is little research on data-driven collections in libraries, especially in physical libraries. Furthermore, there is little implementation of data science methods within physical libraries. There are various theoretical approaches for public libraries, but most implementations involve online book stores, as they have additional (reliable) data, such as user ratings. The features that Lee et al. (2021) used are not available in this data set and in general not available within physical libraries. This is further explained by the fact that public libraries do not have a commercial point of view and lack skilled personnel as Leorke et al. (2018) stated earlier. Concerning data-driven collections this could propose an issue, as books that do not 'sell well' still need to be in a library for cultural and educational purposes (Audunson et al., 2019). The inability to track customer behaviour in detail is another challenge to face, in contrast to online retailers who can track every move a customer makes. This thesis will try to fill the gap of implementation and attention to data science in physical libraries by proposing a technical implementation within public libraries and the Dutch library system.

4 METHODOLOGY AND EXPERIMENTAL SETUP

In this section, the procedures that are used for answering the research questions are further elaborated upon. All discussed procedures are performed per library. The data science pipeline (Figure 1) gives a schematic overview of the workflow that is followed to achieve the desired result. This thesis will achieve the desired result of answering the research question by investigating the relationship between book characteristics, transaction and inventory records. The target variable is constructed from the transaction and inventory records and the book characteristics are used as variables for prediction. The differences in feature importances between libraries will be analysed using the best-performing model.

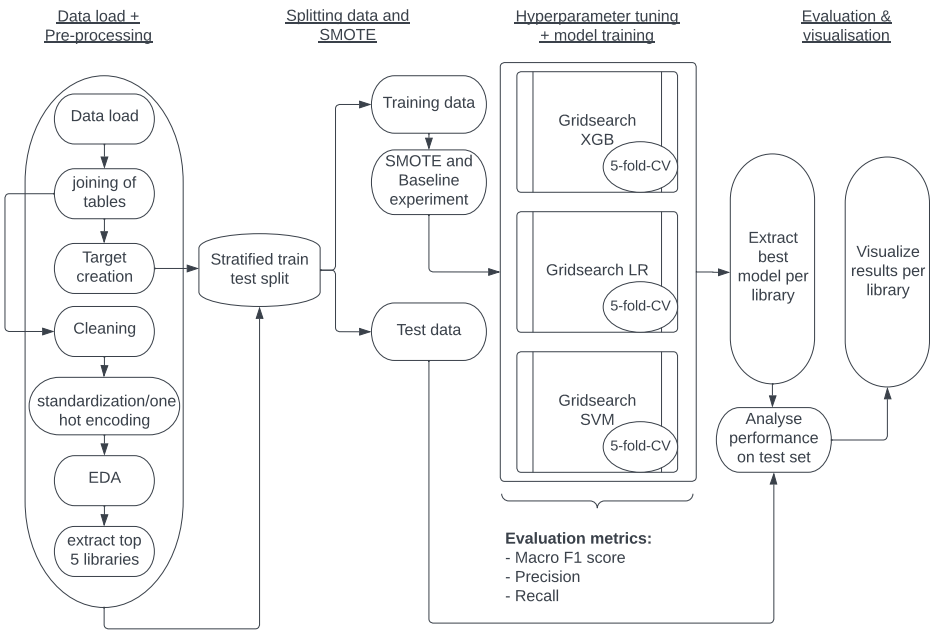

Figure 1: A schematic overview of the data science pipeline used in this thesis.

The data science pipeline (Figure 1) includes various data science techniques such as joining tables, standardization, categorical encoding, splitting of data, oversampling, hyperparameter tuning and evaluation metrics. These techniques all will be explained in chronological order conform Figure 1. The procedures that are explained regarding data cleaning, pre-processing and target creation are performed per library. This results in a total of 82 different data frames, one for each library. Due to computational complexity and time constraints, the hyperparameter optimization, model training and evaluation will only be performed on the biggest five libraries. Resulting in a unique model per library, as the

foremost goal is to predict library-specific popularity scores. All models and libraries are treated equally with the same settings and code, this ensures a comparable result between libraries. To allow for reproducibility a seed has been set.

4.1 *The data set*

The data set is combined from a client and partner of Enjins; NBD Biblion and the KB (Royal Library of the Netherlands). NBD Biblion is distributing over 2 million books a year to various libraries across the Netherlands (NBD Biblion, 2021). The KB keeps track of the collection of each public library. For this thesis the data of 2020 and 2021 is used, as in previous years the data was not complete enough to perform an adequate analysis. NBD Biblion made substantial progress in the registering of data in the past years. The data contains 6,114,042 transactions and 19,057,132 inventory records for a total of 82 libraries. Data from NBD Biblion entails the book characteristics from the books they deliver to their clients. It includes 79,661 unique books with variables like genre, subgenres, reading difficulty, language, educational level, author, age group. Furthermore, it includes the physical dimensions and price of every book, which entails 83,470 unique books. The Ai-week CSV consists of 31,157 unique books, for these books the generic popularity score is available. The raw data is distributed over 5 CSVs. In Table 1, the size of each data set is defined including a description of the data set.

Table 1: Characteristics of the raw data, per CSV.

Data set	**Rows**	**Columns**	**Description**
Inventory	19,057,132	6	Inventory data of every library, measured on multiple dates.
Transactions	6,114,042	7	Transactions of books, two types: reservations and loans.
Book characteristics	79,661	17	Content characteristics of a book.
Physical characteristics	83,470	6	Number of pages, heights, prices of a book.
Ai-week	31,157	9	The generic popularity score, for final manipulation.

4.2 *Cleaning and pre-processing*

The data has been through extensive cleaning and pre-processing to generate a workable data set per library. This includes the joining of different CSVs, categorical encoding, dimensionality reduction and the creation of the target label. At first, the transaction and inventory data sets are joined using an inner join, based on the primary identifier for a book (*PPN*). This inner join ensures that for every book a label can be created, it works that the new data set is the intersection $\cap$ of both data sets and uses Equation 1, where A is the inventory records and B is the transaction records.

$$A \cap B \; on \; A_{ppn} \; and \; B_{ppn} \tag{1}$$

Not all the variables in the book characteristics were used because some variables (*genre/cluster and pim/siso*) explain similar information but on a different hierarchy. In collaboration with NBD Biblion, the choice is made to reduce dimensionality by applying feature selection based on domain knowledge. The variables that are included in the future pre-processing steps are *price, height, N pages, cluster, pim, child/adult, fiction/nonfiction, language and author.* These selected book characteristics will be joined using an inner join, based on *PPN*. As seen in Equation 2, where it is a union $\cup$ between A, the content characteristics and B, the physical characteristics. Ai-week will be kept separately and can be used for calculating the final popularity score within the data-driven collection platform of NBD Biblion.

$$A \cup B \; on \; A_{ppn} \; and \; B_{ppn} \tag{2}$$

With regard to the categorical variables in this data set one hot encoding and frequency encoding is applied. This thesis makes a distinction between normal and high cardinality categories. A high cardinality feature includes values that occur rarely (Moeyersoms & Martens, 2015), an example is the author variable. The number of unique values per category can be found in Table 2. One hot encoding is applied on every categorical variable except the author variable, as it is high in cardinality. This entails the creation of a new column for every unique value per category. This is done to ensure that there are only numerical values in the data set and that each value is treated independently (Fawcett, 2021). Although these categories are not high in cardinality, some values in the category appear very rarely ($< 5\%$ of the instances) for these levels another category called *other* is created to reduce the risk of noise, sparsity and dimensionality (Gilad, 2021). The author variable is frequency encoded into four categories of occurrence, this implies that the value is encoded using the frequency distribution of every unique value and helps the model understand this variable without increasing the dimensionality drastically (Singh, 2018).

Table 2: Unique levels per categorical variable.

Category	PIM	Cluster	Language	Fiction/Non fiction	Adult /Child	Author
N unique Levels	27	9	11	2	2	11,901

It is recommended to use a standardization technique on numeric variables, as it leads to an increase in the overall performance of the model (Luor, 2015). The numeric variables (*price, N pages and height*) will be standardized, by removing the mean μ from the data point X and scaling it to a unit variance σ as seen in Equation 3. This is applied to the data as those variables have a broad range of values, which results in large distances between data points.

$$z = \frac{(X - \mu)}{\sigma} \tag{3}$$

4.3 *Target label construction*

Using the joined transaction and inventory data frames, the target label is created. This is created as there is no pre-existing target label or user ratings that explains how popular a book is in a library compared to other libraries. The steps to achieve this target label are explained below. To further demonstrate these procedures, an example of the steps is shown in Appendix B.

The transaction data has a variable called *type*. This indicates the kind of transaction it is, namely a reservation or a loan (Appendix B.1). Domain knowledge at NBD Biblion state that reservations are more important than loans, as customers especially want that book. Therefore, the transactions which have this type are multiplied by 1.5, as agreed upon with NBD Biblion (Appendix B.3). The transaction records are then grouped and summed per book and library. Afterwards, they are divided by the inventory records of that book (Appendix B.2). This creates a *transaction inventory ratio* (Appendix B.4). This ratio is then summed across libraries per book, as it is essential to compare book performance across the same books and not different books in the same library. This value is divided by the *transaction inventory ratio* to get the influence of a book compared to that book in other libraries. This variable is called the *inter ratio* (Appendix B.5). To reduce the effect of outliers, the Robust Scaler of Scikit-learn[3] is applied to scale the score of a book across libraries between the first and third quantile of that book (Appendix B.6). The Robust Scaler removes the

[3] Scikit-learn is a machine learning library, that allows for implementing machine learning models and includes pre-processing methods such as the Robust Scaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Median from data point X and scales the data in the interquartile range (P_{75}-P_{25}), as seen in Equation 4 (Scikit-learn, n.d.). This results in a distribution per book where the effect of outliers is minimized which is beneficial for the model (Jun, Chang, & Jun, 2020).

$$X = \frac{(X - Median)}{P_{75} - P_{25}} \tag{4}$$

To conclude the label creation, the values are binned into a category to get control over the effect the predicted value will have on the generic popularity score. The number of classes chosen is four, the desired amount by NBD Biblion. The class indicates how good or bad a book performs in a library. Thus, it is important to get equal bin sizes above and below zero and standardize this across libraries. These bin sizes ensure that the influence of a predicted label of a book is explainable across libraries, as the bins follow a similar pattern. Therefore, it is not possible to create custom bin sizes for each library, such that in every bin an equal amount of data instances are present. The categories with their influence are listed in Appendix C. The bins and thus the categories used are as follows: *[-inf, -0.2, 0, 0.2, inf]*, *[0, 1, 2, 3]*.

By choosing the specified binning technique which ensures equal bins across libraries, class imbalance is introduced in the target label. Class imbalance indicates that the distribution is not evenly spread across classes. This could cause substantial performance loss, as machine learning models are not able to learn the properties of the minority class in its entirety. As a result, a bias is introduced to the majority class by wrongly classifying the minority class (Guo, Yin, Dong, Yang, & Zhou, 2008). To combat this problem, it is possible to over-or undersample the data set to ensure equal classes. Oversampling of the data set can be achieved with a technique called random oversampling, which randomly duplicates instances in the training set. This is also its downside as it might create too many duplicates and does not represent the actual class properties well (Shamsudin, Yusof, Jayalakshmi, & Akmal Khalid, 2020). To prevent this, another technique called Synthetic Minority Oversampling Technique (SMOTE) was implemented. This technique oversamples the minority classes by creating synthetic samples based on K nearest neighbours. SMOTE is designed by Chawla, Bowyer, Hall, and Kegelmeyer (2002) as a method to combat class imbalance. SMOTE is only applied on the training data, with parameter K set to five, and not on the test data. Oversampling the test data could make the model biased and overoptimistic in classifying (Santos, Soares, Abreu, Araujo, & Santos, 2018). To allow for a baseline, the models will also be evaluated upon the original training set. This provides a more profound conclusion in determining if SMOTE is working as intended. To

create more robust results all experiments are performed five times and the average will be taken and reported. The class distribution of the training set before and after SMOTE is demonstrated for one library (Figure 2 and Figure 3). The distribution of the other libraries are listed in Appendix D as all libraries have imbalanced classes. The exact number of data points manipulated by SMOTE can be found in Appendix E.

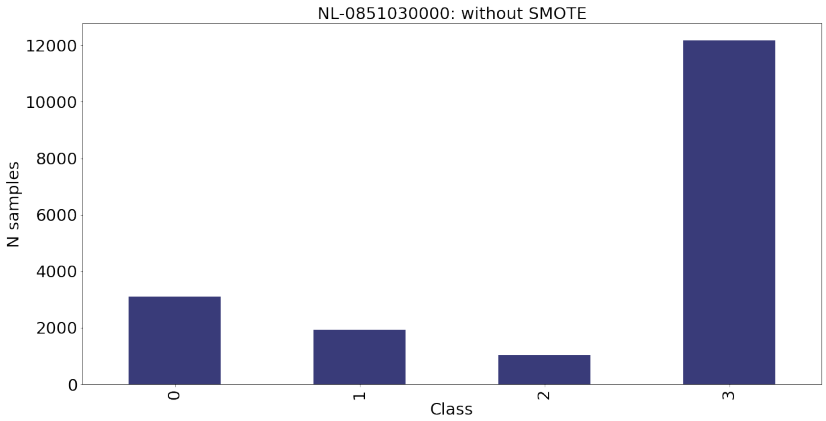

Figure 2: Distribution of the target label within library NL-0851030000, before SMOTE is applied.

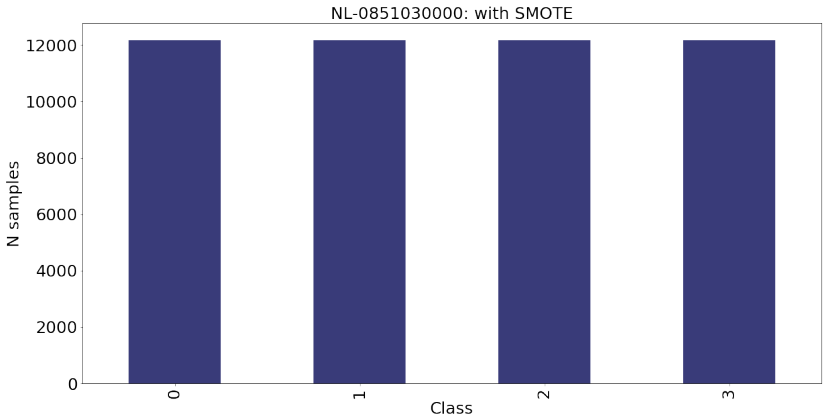

Figure 3: Distribution of the target label within of library NL-0851030000, when SMOTE is applied.

4.4 *Model description and tuning*

Due to the presence of target labels, this is a supervised machine learning problem. As the labels are classes, this is a classification problem. For classification, many machine learning algorithms can be used. As elaborated upon in Section 3, this thesis will implement a XGB, SVM and LR model.

4.4.1 *XGB, SVM and LR*

Boosting is an effective method to produce an accurate prediction rule by combining several weak rules, also called weak learners that are generated at each iteration. The final combined rule is called the strong learner (Schapire, 2003). XGB was proposed by Chen and Guestrin (2016) and is a variant of boosting, namely gradient boosting. Gradient boosting differs from traditional boosting in the sense that the errors are minimized by a gradient descent algorithm. XGB is the primary model, as it is proven to work well with tabular and structured data that includes many categorical variables (Shwartz-Ziv & Armon, 2022). XGB is also explainable and understandable, as feature importance and the final decision tree can be extracted. It is scalable, flexible, sparsity aware, combats overfitting and supports missing values (Bentéjac, Csörgő, & Martínez-Muñoz, 2020). XGB is a decision tree ensemble that generates a weak learner (a decision tree) at each iteration and updates the next decision tree based on the outcome. Using these combined trees, the XGB algorithm creates accurate prediction rules. The sparsity awareness feature is apprehended in the algorithm by removing these entries from the calculation of the gain in the splitting of the tree candidates. This results in a faster algorithm than regular gradient boosting (Bentéjac et al., 2020).

To combat overfitting, the XGB algorithm has two parts in the objective function: the loss function and the regularization term. These can be set to LASSO and Ridge regression, also referred to as L1, L2 (Pesantez-Narvaez, Guillen, & Alcañiz, 2019). The objective function is set to Multi:Softmax, which makes the algorithm capable of handling multi-class classification problems by outputting a probability of each data point belonging to a certain class (XGBoost, 2021a). The evaluation metric used within XGB is the *mlogloss*, which calculates the log loss for a multi-class problems. This is defined in Equation 5, where M is the number of classes, log is the natural logarithmic, y is the indicator if a classification is correct and $\mathbf{P}$ is the predicted probability of class c.

$$-\sum c = 1M\ yo, clog(\mathbf{P}o, c) \tag{5}$$

Furthermore, XGB made remarkable improvements regarding implementation by including parallelized learning, out-of-core computing and cache awareness (Tanha, Abdi, Samadi, Razzaghi, & Asadpour, 2020). It also includes tree pruning and allows for the use of Graphical Processing Units (GPU), which results in fast and accurate models (Mitchell & Frank, 2017).

The GPUs of Google Colab Pro[4] are applied to ensure faster processing and utilize the GPU-XGB model. Feature importances can be extracted via the XGB model which helps with understanding the models choices in detail. Furthermore, this is needed to examine if there are differences between libraries. Variants of feature importance are *gain, weight, cover, total gain and total cover* (XGBoost, 2021b). This thesis uses *gain*, as it indicates the average gain across all splits and entails accuracy improvements that the variable is representing across the tree branches (Yasodhara, Asgarian, Huang, & Sobhani, 2021).

The other models used in this thesis are the SVM and LR classifiers. SVMs are widely used in machine learning for classification tasks, as they are generalizable and have a good performance in real-world scenarios (Yue, Li, & Hao, 2003). When moving from linear kernels towards nonlinear kernels and working with high dimensional data, the computational complexity increases. Working with traditional SVMs could become infeasible (Sadrfaridpour, Razzaghi, & Safro, 2019). To combat this problem, the Thunder-SVM package, proposed by Wen, Shi, Li, Bingsheng, and Chen (2018), is implemented. This allows the use of GPUs and for a faster processing of SVM and provides an excellent balance between performance and quality (Alshanik, Apon, Herzog, Safro, & Sybrandt, 2020). The LR is a popular classifier that uses a logistic function, mostly used for binary classification. It can however be extended to a multi-class classifier, then called a multinomial logistic regression (Boateng & Abaye, 2019). Equation 6 illustrates a one versus rest approach, which entails that multiple binary classifiers C are created. Then each classifier f_c compares a single class to all other classes on X and the highest score for a single class is chosen (Yeh, 2018).

$$\hat{y} = \arg max_{c \in \{1,\dots,C\}} f_c(x) \tag{6}$$

[4] Google Colab is an online framework where users can make use of the GPU from Google within their Python script. Google Colab Pro, is a paid version, which allows for better GPUs.

4.4.2 *hyperparameter tuning*

Splitting of the data is achieved by using a stratified approach. This approach uses a split distribution of 75% and 25% for the training and test set. This ensures that the train and test set hold the same distribution in labels. This is beneficial for model performance and generalizability as there is a similar amount of data for a label present in both training and test sets (Sechidis, Tsoumakas, & Vlahavas, 2011).

To optimize these models, a method called Grid Search[5] is performed on the training set. Grid Search finds the optimal hyperparameters per model and library. It does so by searching the specified subset of hyperparameters provided and finding the best combination for the model. Its advantage is that it always finds the best combination of parameters. However, the downside is that it is computationally expensive in comparison to a randomized search across these parameters (Aryo Anggoro & Sasmita Mukti, 2021). The hyperparameters that are included in the Grid Search are found in Table 3. It uses five-fold cross-validation, which ensures a more robust performance, as the model is trained and evaluated on multiple parts of the training set instead of only relying on the train-test split at the start (Soper, 2021). Overfitting is a problem in nonlinear machine learning algorithms, especially when the algorithm is complex with respect to the number of instances (Bilger & Manning, 2013). Consequently, this applies to XGB. Therefore, an early stopping criterion was implemented to prevent overfitting. This stops the model from training when the performance has not increased in N rounds.

Table 3: hyperparameters per model that is used in Grid Search.

XGBOOST		**SVM**		**LR**	
hyperparameter	Value	hyperparameter	Value	hyperparameter	Value
Max depth	5, 10, 25	C	0.1, 1, 10	Solver	newton-cg, saga, lbfgs
Eta	0.01, 0.1	Gamma	0.1, 0.01	Penalty	None, l2
N estimators	100, 300, 500	Kernel	Linear, poly, rbf, sigmoid	C	0.001, 0.01, 1
Colsample by tree	0.6, 0.8			Max iter	500, 1000, 3000
Gamma	0.5, 1				
Subsample	0.6, 0.8				

[5] Grid Search is a hyperparameter tuning technique from Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

4.5 *Evaluation metrics*

To evaluate the models on the test set, the following evaluation metrics are used: precision, recall and (Macro) F1 score. In the equations below, the formulas are described for all used evaluation metrics. All metrics are calculated using confusion matrices. In these matrices, the TP (true positives), FN (false negatives), FP (false positives) and TN (true negatives) can be obtained to calculate the metrics below.

Precision (Equation 7) tells how precise the model is (per class). This is beneficial to extract insights into the performance per class and if the model can be trusted in predicting certain classes.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall (Equation 8) indicates missed positive predictions. Precision and recall can both be calculated per class.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

As precision and recall both have a different position, the F1 score (Equation 9) is used to calculate the Macro F1 score (Equation 10). The Macro F1 score is an evaluation metric that provides the harmonic mean between precision and recall and treats all classes equally by taking the average of the class-specific F1 scores.

$$F1\ score\ per\ class = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

$$Macro\ F1 = \frac{\sum F1\ score\ per\ class}{N\ classes} \tag{10}$$

4.6 *Implementation and packages*

The code used in this thesis is written in Python (Rossum, G, D, & Fred, 2009). This is chosen as it is the preferred programming language for scientific computing, machine learning and data science, due to performance and low-level APIs (Raschka, Patterson, & Nolet, 2020). Packages, versions and external programs that are used in this thesis can be found in Table 4, where the use case is also described.

Table 4: Packages, external programs, their versions and use.

Program or package	**Version**	**Use**
Python	3.85	Interpreter
Numpy	1.19.5	Calculations
Pandas	1.3.2	Importing data, manipulating data
Scikit-Learn	1.0.2	Machine learning models, grid search, scaling
Matplotlib	3.3.2	Plotting
Pickle	0.7.5	Saving dataset, saving models, saving the output of grid search
Imblearn	0.9.0	SMOTE
Google Colab Pro	N.A	Working with GPUs to speed up computational time

5 RESULTS

In this section, the multi-class classification performance of four classes is examined. Therefore, the performance of the XGB, SVM and LR models concerning the prediction of library-specific book popularity scores are explained per library. These performances are accomplished after hyper-parameter tuning. The results of this and consequently the parameters every model has used can be found in Appendix F. Each of the models within a library is evaluated using the test set. This ensures that all models within a library are evaluated on the set. The test set was kept apart during training and is thus not manipulated by SMOTE. The results presented in this section relate to the five libraries that were chosen earlier in this thesis. The results are demonstrated per evaluation metric in the following order: NL-0800070000, NL-0800790000, NL-0830860000, NL-0851030000 and NL-0873440000.

First, the results of all models and libraries that were trained upon the original training set are described. This is used as a baseline (Section 5.1). As mentioned before, the original training set has not been modified with SMOTE. This experiment will be named the Baseline-experiment and the models within are named as follows: XGB-baseline, SVM-baseline and LR-baseline. Section 5.2 investigates the results of all models that were trained upon the training set that is oversampled by SMOTE. This experiment will be named the SMOTE-experiment and the models within the experiment are named as follows: XGB-SMOTE, SVM-SMOTE and LR-SMOTE. This will be followed by a comparison between the baseline- and SMOTE-experiments (Section 5.3). At last, the feature importances of the best-performing method will be discussed in Section 5.4.

5.1 *Baseline-experiment*

The Macro F1 scores of the XGB-baseline, SVM-baseline and LR-baseline models can be found in Figure 4. The table with the corresponding values is listed in Appendix G, Table 22. Figure 4 shows that XGB-baseline and SVM-baseline are the best performing models across libraries with Macro F1 score of 0.35, 0.31, 0.33, 0.29 and 0.33 for the XGB-baseline model and 0.29, 0.29, 0.30, 0.33 and 0.33 for the SVM-baseline model. LR-baseline performs the worst in all libraries (except for libraries NL-080079000 and NL-0830860000) with Macro F1 scores of 0.22, 0.30, 0.32, 0.27 and 0.18 respectively.

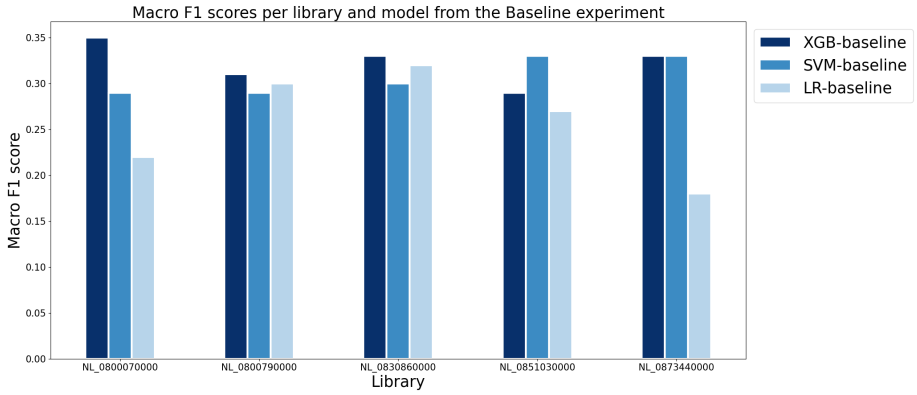

Figure 4: Macro F1 scores per model and library in the Baseline-experiment.

To investigate these results more thoroughly, the between-class performance is described. This is done using class-specific precision and recall, which can be found in Tables 5 and 6. Here, the highest measured precision and recall values are highlighted in bold. The precision and recall per class show remarkable high values for class 3 in libraries NL-0830860000, NL-0851030000 and NL-0873440000. When looking at the distribution of the target label in Appendix D. It is shown that class 3 is the majority class within those libraries. Class 0 is performing relatively well with respect to classes 1 and 2. From Appendix D it is also shown that this class is the majority class in two libraries (NL-0800070000, NL-0800790000) or the second predominant class in the other libraries. Notably, there are several 0.00 and 1.00 values for both the precision and recall. These values all lie in the minority class and the majority class respectively, as shown in Appendix D. This pattern applies especially to the SVM-baseline and LR-baseline models.

Table 5: Precision per class (0, 1, 2, 3) and model within the Baseline-experiment. The highest performing class within a library and model is marked in bold.

Library	Precision per class											
	XGB-baseline				SVM-baseline				LR-baseline			
	0	1	2	3	0	1	2	3	0	1	2	3
NL-0800070000	**0.54**	0.23	0.22	0.40	**0.51**	0.25	0.00	0.42	**0.50**	0.00	0.00	0.40
NL-0800790000	**0.50**	0.26	0.05	0.37	**0.47**	0.33	0.00	0.44	**0.46**	0.39	0.00	0.37
NL-0830860000	0.27	0.22	0.16	**0.69**	0.40	0.16	0.00	**0.66**	0.50	0.20	0.00	**0.65**
NL-0851030000	0.27	0.13	0.06	**0.69**	0.35	0.33	0.00	**0.67**	0.42	0.00	0.00	**0.67**
NL-0873440000	0.27	0.20	0.11	**0.73**	**0.75**	0.00	0.00	0.72	0.00	0.00	0.00	**0.72**

Table 6: Recall per class (0, 1, 2, 3) and model within the Baseline-experiment. The highest performing class within a library and model is marked in bold.

Library	Recall per class											
	XGB-baseline				SVM-baseline				LR-baseline			
	0	1	2	3	0	1	2	3	0	1	2	3
NL-0800070000	**0.69**	0.15	0.05	0.37	**0.87**	0.00	0.00	0.28	**0.90**	0.00	0.00	0.20
NL-0800790000	**0.66**	0.17	0.01	0.34	**0.93**	0.03	0.00	0.15	**0.93**	0.02	0.00	0.11
NL-0830860000	0.18	0.10	0.03	**0.87**	0.03	0.01	0.00	**0.99**	0.26	0.18	0.07	**1.00**
NL-0851030000	0.16	0.04	0.01	**0.89**	0.02	0.00	0.00	**0.99**	0.01	0.00	0.00	**1.00**
NL-0873440000	0.10	0.05	0.02	**0.94**	0.01	0.00	0.00	**1.00**	0.00	0.00	0.00	**1.00**

5.2 *SMOTE-experiment*

Using the Macro-F1 scores in Figure 5, the table with the corresponding values are listed in Appendix G, Table 23. The highest scores are yielded by the XGB-SMOTE algorithm, which are 0.33, 0.30, 0.32, 0.31 and 0.29. The SVM-SMOTE and LR-SMOTE Macro F1 scores are 0.27, 0.27, 0.32, 0.27, 0.22 and 0.25, 0.28, 0.31, 0.29, 0.24 respectively.

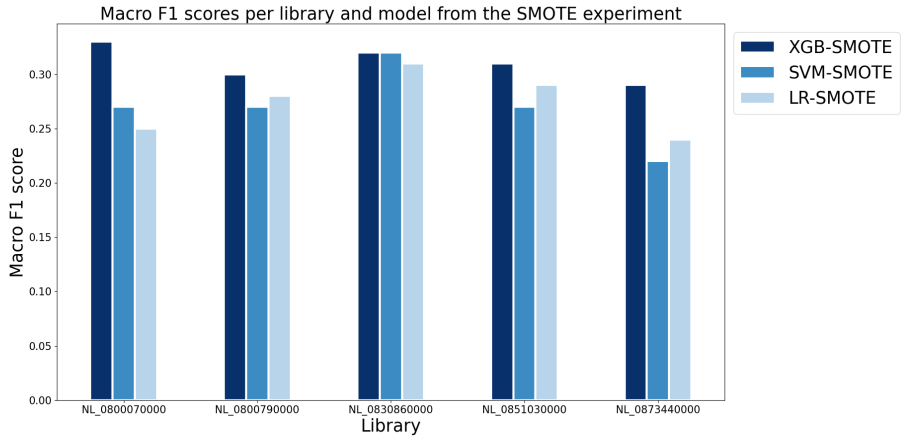

Figure 5: Macro F1 scores per model and library in the SMOTE-experiment.

To analyse the performance thoroughly, the between-class performance is evaluated. The precision and recall per class can be found in Tables 7 and 8, where the highest values are marked in bold. Noticeably, the performance in class 3 is exceptionally high for three out of five libraries (NL-0873440000, NL-0851030000 and NL-0830860000). Interestingly, this pattern applies to all models. Additionally, class 2 is consistently performing the lowest in all libraries. When looking at Appendix D, it shows that within those libraries the class was particularly imbalanced towards class 3 and in all libraries class 2 is the least present class. In the other libraries

(NL-0800070000, NL-0800790000), the class imbalance is less severe and class 0 was the most predominant class. However, the same pattern holds, as in those libraries the majority class is the best performing class.

Table 7: Precision per class (0, 1, 2, 3) and model within the SMOTE-experiment. The highest performing class within a library and model is marked in bold.

Library	Precision per class											
	XGB-SMOTE				SVM-SMOTE				LR-SMOTE			
	0	1	2	3	0	1	2	3	0	1	2	3
NL-0800070000	**0.56**	0.26	0.12	0.39	**0.63**	0.23	0.06	0.38	**0.55**	0.24	0.06	0.35
NL-0800790000	**0.51**	0.24	0.11	0.34	**0.54**	0.23	0.08	0.35	**0.54**	0.23	0.05	0.34
NL-0830860000	0.27	0.22	0.10	**0.71**	0.26	0.20	0.08	**0.77**	0.26	0.18	0.07	**0.75**
NL-0851030000	0.29	0.12	0.12	**0.71**	0.26	0.12	0.07	**0.76**	0.25	0.14	0.07	**0.76**
NL-0873440000	0.22	0.15	0.07	**0.75**	0.15	0.09	0.07	**0.74**	0.17	0.12	0.07	**0.75**

Table 8: Recall per class (0, 1, 2, 3) and model within the SMOTE-experiment. The highest performing class within a library and model is marked in bold.

Library	Recall per class											
	XGB-SMOTE				SVM-SMOTE				LR-SMOTE			
	0	**1**	**2**	**3**	**0**	**1**	**2**	**3**	**0**	**1**	**2**	**3**
NL-0800070000	**0.58**	0.23	0.11	0.40	0.31	0.15	0.42	**0.40**	0.35	0.05	0.44	**0.39**
NL-0800790000	**0.54**	0.23	0.11	0.33	0.35	0.18	**0.40**	0.30	**0.46**	0.20	0.11	0.37
NL-0830860000	0.25	0.20	0.10	**0.74**	0.38	0.16	0.17	**0.62**	0.46	0.12	0.04	**0.69**
NL-0851030000	0.28	0.10	0.09	**0.75**	0.39	0.20	0.28	**0.40**	0.47	0.14	0.18	**0.50**
NL-0873440000	0.20	0.12	0.06	**0.79**	0.23	0.15	0.34	**0.33**	0.29	0.05	0.32	**0.47**

5.3 *Baseline- and SMOTE-experiment comparison*

When comparing the performance of the models that are trained on the original training set (Baseline-experiment) and the oversampled training set by SMOTE (SMOTE-experiment) as seen in Figure 6. It shows that the XGB-baseline model is outperforming the XGB-SMOTE model in four out of five libraries. Regarding the SVM, the same pattern applies. The LR deviates from this pattern, as it is the least performing predictor. However, LR-SMOTE models outperform the LR-baseline in three out of five libraries. Looking at the class-specific precision, recall and overall Macro F1 scores, the XGB-baseline model is yielding the highest overall performance. Additionally, it generalizes the best to minority classes with respect to the SVM-baseline and LR-baseline. In Section 5.4, only the XGB-Baseline model is evaluated on its feature importance. This model performs the best and will therefore be further analysed to get a deeper understanding of the variables that the model finds important.

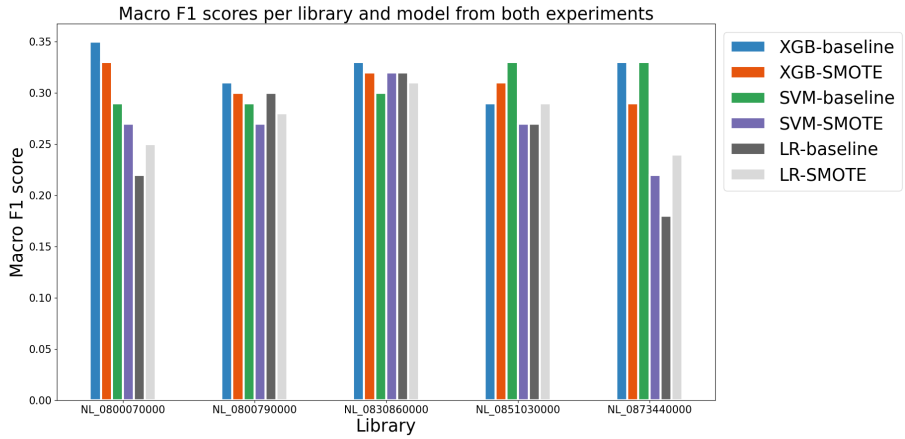

Figure 6: Macro F1 scores per model and library from the Baseline and SMOTE-experiment.

5.4 *Feature importances*

To analyse the variables that the XGB-baseline model prioritizes between libraries, feature importances are found in Appendix H. These demonstrate the variables that the XGB-baseline model finds most important given the data. Noticeably, the top three important variables are shared between all libraries, which are *number of pages, price and height*. Additionally, the variable *language code other* is consistently the least influential predictor across the libraries. The content variables that represent specific genres, appear in the middle of the feature importances in all libraries. There are no obvious results that demonstrate differences between libraries that had different high-performing classes (as outlined in Section 5.1) such as libraries NL-0800070000, NL-0800790000 and libraries NL-0830860000, NL-0851030000, NL-0873440000.

6 DISCUSSION

The goal of this thesis is to realize library-specific data-driven book popularity scores, which could then be used by NBD Biblion to manipulate the generic popularity scores. Furthermore, the variables were analysed using feature importances to see if there are notable differences between libraries. By proposing a technical implementation, this method aims to fill up the lack of implementation and attention to data science in physical libraries. Sections 6.1, 6.2, 6.3 and 6.4 will evaluate the results and refer back to the research questions that were stated in Section 2. In Section 6.5, the results are discussed in more depth concerning the research questions. Section 6.6 will outline suggestions which can be used in future work.

6.1 *Oversampling*

Before sub-question one can be addressed, the implications of oversampling with SMOTE are discussed. SMOTE is applied to resolve the class imbalance as, this could improve generalizability and performance. From the overall results, it is obvious that using SMOTE does not contribute notably to facilitating higher performance. It is even decreasing performance on the test set in certain cases. It can thus be stated that oversampling the training set using SMOTE did not have the expected effect on the performance concerning the test set (which was not oversampled by SMOTE). This holds for both XGB and SVM models. Concerning the LR model, it was indecisive if SMOTE contributed or not, as the LR-SMOTE models only performed better than the LR-baseline models in certain libraries.

6.2 *Model comparison*

With regard to sub-question one, it can be said that the XGB-baseline model is the best performing model across all classes and libraries (with an exception of one library). Furthermore, it is the most generalizable when comparing it to the SVM and LR models in the baseline experiment. The overall model performance was notably lower than Lee et al. (2021) achieved in their study, but the models did show predicting power within the majority classes. It can thus be stated that the between-class analysis unveils various problems concerning the distribution of the target variable.

6.3 *Feature importances*

With regard to sub-question two, feature importances from the XGB-baseline model show that numerical values are thriving the predictions in all libraries in contrast to other (content) variables. The foremost difference between those variables is the fact that content variables are one hot encoded and thus the impact a single variable has is spread out. It is therefore apparent that this is not the best method to create insights between libraries. Additionally, These feature importances did not create further insights between libraries and their different distributions.

6.4 *Main research question*

With respect to the main research question, it can be said that book characteristics can explain the lending behaviour of a library when looking at class-specific performance. However, the high-performing classes were trained upon sufficient real instances. This demonstrates problems concerning the target distribution before model training. It can be said that there are various problems to solve before the proposed method is implementable within the Dutch library system.

6.5 *Error analysis and limitations*

The results vary when looking at the between-class performance from the Baseline-experiment. As the majority classes looked particularly promising, as those classes outperformed the other classes in all libraries. Concerning the poor-performing classes, it is clear that there were insufficient data points to learn the class properties. Looking at the between-class performance concerning the SMOTE-experiment and Baseline-experiment, a similar pattern is unveiled. Classes that were trained on a combination of artificial instances generated by SMOTE and the original data points did not represent the class properties enough that the test set could be correctly predicted. Therefore, it is likely that the artificial instances do not reflect the class properties well when seeing actual data points from the test set. This suspicion is confirmed by Anis and Ali (2017), who found that a substantial class imbalance ratio within a classification problem could cause bad performance when using SMOTE. Anis and Ali (2017) further state that the model becomes extremely biased when dealing with a large imbalance ratio, which in turn decreases model performance. This is also confirmed by Wentao, Wang, and Wang (2015) who declared that SMOTE (when suffering from severe imbalance) does no longer satisfy the distribution of the variables resulting in a weaker classification.

This phenomenon is further supported in this thesis, as class 2 is the least common class in each library and model. It also predicted the poorest in both the Baseline-experiment and SMOTE-experiment. The between-class performance explains the relatively low Macro F1 scores. As within the SMOTE-experiment the instances in the training set are not representative of the real instances in the test set. Thus when predicting on the test set the instances are wrongly classified. This is further verified by the precision and recall per class. Here, the classes with the highest number of SMOTE instances in the training set are predicted the worst and vice versa.

Unfortunately, the SMOTE-experiment did not resolve the problem of class imbalance and made performance poorer in certain cases when comparing it to the Baseline-experiment. Therefore, the construction of the target label has to be discussed. This thesis aimed for four classes. However, this could be changed to fewer classes, to resolve the class imbalance naturally. However, this has the downside of decreasing the information this score holds. Another solution could be to change the used scaler. In this thesis, the Robust Scaler was used, but alternatives could be considered which, besides handling outliers, scale the data more evenly. Additionally, classes could be removed in general by using regression analysis. Nevertheless, this could introduce new challenges, such as the interpretability of the score and raises questions with regard in controlling the effect the predicted score has on the general popularity score.

As discussed above, the limitations that most influence the performance of the models in both experiments are related to the target variable and its construction. This is based upon an in-depth analysis of class performances. SMOTE showed limitations in generating sufficient reliable instances that represent the real instances in the test set well when comparing it to the Baseline-experiment. The target construction could be altered to improve the imbalance and the overall performance. This should be done thoughtfully as the bin sizes have to be the same across the libraries.

Feature importances show that numeric variables are the dominant predictors compared to the one hot encoded variables. This could be caused by the high sparsity of the data set, as all categorical variables are one hot encoded. Additionally, the effect a single level of a categorical variable has on a prediction is much lower than all levels of that category combined, thus both variable types are hard to compare with each other. Furthermore, the variable *language code other* is always the least influential predictor. This is explained by the fact that it is very rare and only occurs in less than 1% of the instances. The feature importances provided by XGB is inadequate in identifying differences within book characteristics across libraries. The

rising importance of eXplainable AI (XAI) could be implemented here. Research by Saarela and Jauhiainen (2021) can be used as a starting point for resolving this issue. They state that it is essential to combine multiple explainability methods to achieve the best insights into the influence that variables have in predicting.

6.6 *Future work*

The contribution of this thesis fits into the workflow that Otterlo (2017) suggested. Even though there are caveats to making this method implementable in all libraries, this thesis made substantial progress in the understanding of library data and implementing a data science application in the public library system. Directions future work could take is that it should be investigated if libraries can collect user ratings about books from their members. According to Lee et al. (2021), this would drastically improve performance. Furthermore, other over-or undersampling techniques should be investigated, as SMOTE showed limitations in creating reliable instances that reflect class properties accurately. Suggested methods are Borderline-SMOTE and Safe-SMOTE. These methods are extensions of SMOTE, in which Borderline-SMOTE only oversamples the minority classes that are near the borderline of a different class (Han, Wang, & Mao, 2005). Safe-SMOTE carefully oversamples the minority classes and only creates synthetic samples if it is considered safe (Bunkhumpornpat, Sinapiromsaran, & Lursinsap, 2009). Undersampling technique Near-Miss should be tested, as it ensures a well-constructed class distribution boundary and preserves the nearest neighbours from the minority class with respect to the majority class (Dennis, Budianto, Azaria, & Gunawan, 2022). Additionally, when data gathering such as collecting user ratings causes problems and the data set does not improve, another method for target creation should be investigated to ensure the balanced classes. In this case, an oversampling technique would no longer be needed.

Another approach for future research is to predict using probabilities, which would result in a directly implementable model. Here, a prediction is accepted or declined based on the confidence of the model. If the model is confident enough, the prediction could be used to manipulate the generic score in the platform of NBD Biblion. Concerning the feature importances within a library, alternatives should be investigated, as XGB does not provide methods that work well with this data set. for instance, an alternative could be including the field of XAI as discussed above.

7 CONCLUSION

This thesis tried to find the answer to the following research question.

> *To what extent can book characteristics in combination with lending behaviour explain library-specific book popularity scores?*

This has been done by answering the following sub-questions.

SQ1 *How accurate can XGBoost predict book popularity scores compared to a Logistic Regression classifier and Support Vector Machine classifier across different libraries?*

SQ2 *How do book characteristics and lending behaviour affect the predictions across libraries?*

To answer the first sub-question, this thesis showed that XGB can predict the popularity scores better than SVM and LR in both experiments. This was expected as Ullah et al. (2021); Çolakoglu and Akkaya (2019) showed similar results in their studies. This conclusion is based upon multiple evaluation metrics and in-depth error analysis on class performance. However, the models from the SMOTE-experiment did not outperform the Baseline-experiment. This underlines that oversampling the training set with SMOTE did not reflect the actual population correctly in the test set. To answer the second sub-question, this thesis showed that numeric and categorical variables notably differ in the prediction power they have within the XGB-baseline model, as numeric variables were predominantly more important than categorical variables. This means that the content categorical variables lack predicting power, as they are one hot encoded. This increases the sparsity and spreads out the impact of the variable. Therefore, this is not the best approach to identify differences regarding content categories between libraries and an alternative should be sought by combining various methods and including the XAI domain.

With these answers, it can be concluded that book characteristics can explain lending behaviour across libraries and XGB is the best performing model across both experiments. However, before this method is implemented in a library, a few caveats should be solved. For instance, sufficient representative data is needed to resolve the class imbalance. Therefore, other over-or undersampling techniques should be investigated. In addition, changing the construction of the target variable could be beneficial. Another solution is to start collecting user ratings or use probabilities in predicting. If these limitations are resolved the proposed method could be implemented in the Dutch library system.

8 ACKNOWLEDGEMENTS

In this section, I would like to thank the people that helped me during the process of writing this thesis. In special, I would like to thank Dr. Görkem Saygili who has been an excellent supervisor in the guiding of this thesis and offered incredible support. Also in special, I would like to thank Sophia Zitman from Enjins, who apart from technical support gave me a warm welcome in the team at Enjins and helped me in any way I needed.

I would also like to thank NBD Biblion, who were kind enough to provide the data set and confidence for this project. Next, I would like to thank my fellow students that gave valuable insights and gave feedback on portions of the thesis. I would also like to thank my family and friends for their great support when I was writing this thesis. At last, I want to thank all the colleagues at Enjins for answering all my questions and providing me with a warm welcome at the office.

Thank you!
Cas Julicher

9 REFERENCES

Aggarwal, C. (2016). *Recommender Systems* (Vol. 1). New York, United States: Springer Publishing. doi: 10.1007/978-3-319-29659-3

Alshanik, F., Apon, A., Herzog, A., Safro, I., & Sybrandt, J. (2020). Accelerating Text Mining Using Domain-Specific Stop Word Lists. *2020 IEEE International Conference on Big Data (Big Data)*, 2639–2648. doi: 10.1109/bigdata50022.2020.9378226

An, Y., Li, H., Su, T., & Wang, Y. (2021). Determining Uncertainties in AI Applications in AEC Sector and their Corresponding Mitigation Strategies. *Automation in Construction*, *131*, 103883. doi: 10.1016/j.autcon.2021.103883

Anis, M., & Ali, M. (2017). Investigating the Performance of Smote for Class Imbalanced Learning: A Case Study of Credit Scoring Datasets. *European Scientific Journal, ESJ*, *13*(33), 340. doi: 10.19044/esj.2017.v13n33p340

Aryo Anggoro, D., & Sasmita Mukti, S. (2021). Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure. *International Journal of Intelligent Engineering and Systems*, *14*(6), 198–207. doi: 10.22266/ijies2021.1231.19

Audunson, R., Aabø, S., Blomgren, R., Evjen, S., Jochumsen, H., Larsen, H., ... Koizumi, M. (2019). Public libraries as an infrastructure for a sustainable public sphere. *Journal of Documentation*, *75*(4), 773–790. doi: 10.1108/jd-10-2018-0157

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2020). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3), 1937–1967. doi: 10.1007/s10462-020-09896-5

Bilger, M., & Manning, W. G. (2013). MEASURING OVERFITTING IN NONLINEAR MODELS: A NEW METHOD AND AN APPLICATION TO HEALTH EXPENDITURES. *Health Economics*, *24*(1), 75–85. doi: 10.1002/hec.3003

Boateng, E. Y., & Abaye, D. A. (2019). A Review of the Logistic Regression Model with Emphasis on Medical Research. *Journal of Data Analysis and Information Processing*, *07*(04), 190–207. doi: 10.4236/jdaip.2019.74012

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In T. Theeramunkong, B. Kijsirikul, N. Cercone, & T.-B. Ho (Eds.), *Advances in knowledge discovery and data mining* (pp. 475–482). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cao, G., Liang, M., & Li, X. (2018). How to make the library smart? The conceptualization of the smart library. *The Electronic Library*, *36*(5), 811–825. doi: 10.1108/el-11-2017-0248

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. doi: 10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. doi: 10.1145/2939672.2939785

Dennis, Budianto, I. R., Azaria, R. K., & Gunawan, A. A. (2022). Machine learning-based approach on dealing with binary classification problem in imbalanced financial data. In *2021 international seminar on machine learning, optimization, and data science (ismode)* (p. 152-156). doi: 10.1109/ISMODE53584.2022.9742834

Fawcett, A. (2021, 02). *Data Science in 5 Minutes: What is One Hot Encoding?* Retrieved from https://www.educative.io/blog/one-hot-encoding

Gilad, M. (2021, 12). *Handling rare categorical values with Pandas | Gett Engineering.* Retrieved from https://medium.com/gett-engineering/handling-rare-categorical-values-in-pandas-d1e3f17475f0

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the Class Imbalance Problem. *2008 Fourth International Conference on Natural Computation*, 129–201. doi: 10.1109/icnc.2008.871

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. In D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Eds.), *Advances in intelligent computing* (pp. 878–887). Berlin, Heidelberg: Springer Berlin Heidelberg.

Hardesty, L. (2022, 01). *The history of Amazon's recommendation algorithm.* Retrieved from https://www.amazon.science/the-history-of-amazons-recommendation-algorithm

Hicks, D., Cavanagh, M. F., & VanScoy, A. (2020). Social network analysis: A methodological approach for understanding public libraries and their communities. *Library Information Science Research*, *42*(3), 101029. doi: 10.1016/j.lisr.2020.101029

Howard, J. (2019). *The Complicated Role of the Modern Public Library.* Retrieved from https://www.neh.gov/article/complicated-role-modern-public-library

Jun, J.-h., Chang, T.-W., & Jun, S. (2020). Quality Prediction and Yield Improvement in Process Manufacturing Based on Data Analytics. *Processes*, *8*(9), 1068. doi: 10.3390/pr8091068

Lee, S., Ji, H., Kim, J., & Park, E. (2021). What books will be your bestseller?

A machine learning approach with Amazon Kindle. *The Electronic Library*, *39*(1), 137–151. doi: 10.1108/el-08-2020-0234

Leorke, D., Wyatt, D., & McQuire, S. (2018). "More than just a library": Public libraries in the 'smart city'. *City, Culture and Society*, *15*, 37–44. doi: 10.1016/j.ccs.2018.05.002

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. doi: 10.1109/mic.2003.1167344

Luor, D.-C. (2015). A comparative assessment of data standardization on support vector machine for classification problems. *Intelligent Data Analysis*, *19*(3), 529–546. doi: 10.3233/ida-150730

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, *3*, 127. doi: 10.7717/peerj-cs.127

Moeyersoms, J., & Martens, D. (2015). Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector. *Decision Support Systems*, *72*, 72–81. doi: 10.1016/j.dss.2015.02.007

NBD Biblion. (2021). *Over NBD Biblion.* Retrieved from https://www.nbdbiblion.nl/over-nbd-biblion

NBD Biblion. (2022, 03). *Reservations loans in public libraries.* Retrieved from https://www.nbdbiblion.nl/

Otterlo, M. (2017). Project BLIIPS: Making the Physical Public Library more Intelligent through Artificial Intelligence. *Qualitative And Quantitative Methods In Libraries*, *5*(2), 287–300.

Palumbo, B. D. (2019, 07). *Amazon at 25: The story of a giant.* Retrieved from https://www.bbc.com/news/business-48884596

Pesantez-Narvaez, J., Guillen, M., & Alcañiz, M. (2019). Predicting Motor Insurance Claims Using Telematics Data—XGBoost versus Logistic Regression. *Risks*, *7*(2), 70. doi: 10.3390/risks7020070

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. *Information*, *11*(4), 193. doi: 10.3390/info11040193

Rossum, V., G, D, & Fred, L. (2009). *Python .* CreateSpace.

Saarela, M., & Jauhiainen, S. (2021). Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, *3*(2), 1–12.

Sadrfaridpour, E., Razzaghi, T., & Safro, I. (2019). Engineering fast multilevel support vector machines. *Machine Learning*, *108*(11), 1879–1917. doi: 10.1007/s10994-019-05800-7

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *2013 International*

Conference on Collaboration Technologies and Systems (CTS), 42–47. doi: 10.1109/cts.2013.6567202

Santos, M. S., Soares, J. P., Abreu, P. H., Araujo, H., & Santos, J. (2018). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]. *IEEE Computational Intelligence Magazine*, *13*(4), 59–76. doi: 10.1109/mci.2018.2866730

Schapire, R. E. (2003). The Boosting Approach to Machine Learning: An Overview. *Nonlinear Estimation and Classification*, 149–171. doi: 10.1007/978-0-387-21579-2\{_}9

Scikit-learn. (n.d.). *sklearn.preprocessing.RobustScaler.* Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

Sechidis, K., Tsoumakas, G., & Vlahavas, I. (2011). On the Stratification of Multi-label Data. *Machine Learning and Knowledge Discovery in Databases*, *6913*, 145–158. doi: 10.1007/978-3-642-23808-6\{_}10

Shamsudin, H., Yusof, U. K., Jayalakshmi, A., & Akmal Khalid, M. N. (2020). Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset. , 803-808. doi: 10.1109/ICCA51439.2020.9264517

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. doi: 10.1016/j.inffus.2021.11.011

Singh, N. (2018, 12). *Overview of Encoding Methodologies.* Retrieved from https://www.datacamp.com/community/tutorials/encoding-methodologies

Skilton, M., & Hovsepian, F. (2017). *The 4th Industrial Revolution*. doi: 10.1007/978-3-319-62479-2

Soper, D. S. (2021). Greed Is Good: Rapid Hyperparameter Optimization and Model Selection Using Greedy k-Fold Cross Validation. *Electronics*, *10*(16), 1973. doi: 10.3390/electronics10161973

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, *7*(1). doi: 10.1186/s40537-020-00349-y

Teets, M., & Goldner, M. (2013). Libraries' Role in Curating and Exposing Big Data. *Future Internet*, *5*(3), 429–438. doi: 10.3390/fi5030429

Thiebaut, R. (2019). Chapter 9 AI Revolution: How Data Can Identify and Shape Consumer Behavior in Ecommerce. *Entrepreneurship and Development in the 21st Century*, 191–229. doi: 10.1108/978-1-78973-233-720191012

Tsuji, K., Kuroo, E., Sato, S., Ikeuchi, U., Ikeuchi, A., Yoshikane, F., &

Itsumura, H. (2012). Use of Library Loan Records for Book Recommendation. *2012 IIAI International Conference on Advanced Applied Informatics*, 30–35. doi: 10.1109/iiai-aai.2012.16

Ullah, H., Ahmad, B., Sana, I., Sattar, A., Khan, A., Akbar, S., & Asghar, M. Z. (2021). Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews. *CAAI Transactions on Intelligence Technology*, *6*(3), 251–264. doi: 10.1049/cit2.12046

Wen, Z., Shi, J., Li, Q., Bingsheng, H., & Chen, J. (2018). ThunderSVM: a fast SVM library on GPUs and CPUs. *The Journal of Machine Learning Research*, *19*(1), 797–801.

Wentao, M., Wang, J., & Wang, L. (2015). Online sequential classification of imbalanced data by combining extreme learning machine and improved SMOTE algorithm. *2015 International Joint Conference on Neural Networks (IJCNN)*, 1–8. doi: 10.1109/ijcnn.2015.7280620

XGBoost. (2021a). *Python API Reference — xgboost 2.0.0-dev documentation.* Retrieved from https://xgboost.readthedocs.io/en/latest/python/python_api.html

XGBoost. (2021b). *XGBoost Parameters — xgboost 2.0.0-dev documentation.* Retrieved from https://xgboost.readthedocs.io/en/latest/parameter.html#learning-task-parameters

Yasodhara, A., Asgarian, A., Huang, D., & Sobhani, P. (2021). On the Trustworthiness of Tree Ensemble Explainability Methods. *Lecture Notes in Computer Science*, *12844*, 293–308. doi: 10.1007/978-3-030-84060-0\{_}19

Yeh, C. (2018, 06). *Binary vs. Multi-Class Logistic Regression* . Retrieved from https://chrisyeh96.github.io/2018/06/11/logistic-regression.html#:%7E:text=Multinomial%20Logistic%20Regression%20(via%20Cross%2DEntropy),-The%20multi%2Dclass&text=Model%3A%20For%20an%20example%20%2C%20the,%E2%81%A1%20max%20i%20z%20i%20.

Yue, S., Li, P., & Hao, P. (2003). SVM classification:Its contents and challenges. *Applied Mathematics-A Journal of Chinese Universities*, *18*(3), 332–342. doi: 10.1007/s11766-003-0059-5

Zhan, M., & Widén, G. (2018). Public libraries: roles in Big Data. *The Electronic Library*, *36*(1), 133–145. doi: 10.1108/el-06-2016-0134

Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, *21*(6), 1487–1524. doi: 10.3233/ida-163209

Çolakoglu, N., & Akkaya, B. (2019). Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases. *Recent Advances n Data Science and Business Analytics*, *2019*, 162.

10 APPENDICES AND SUPPLEMENTARY MATERIALS

A *Current implementation of generic score*

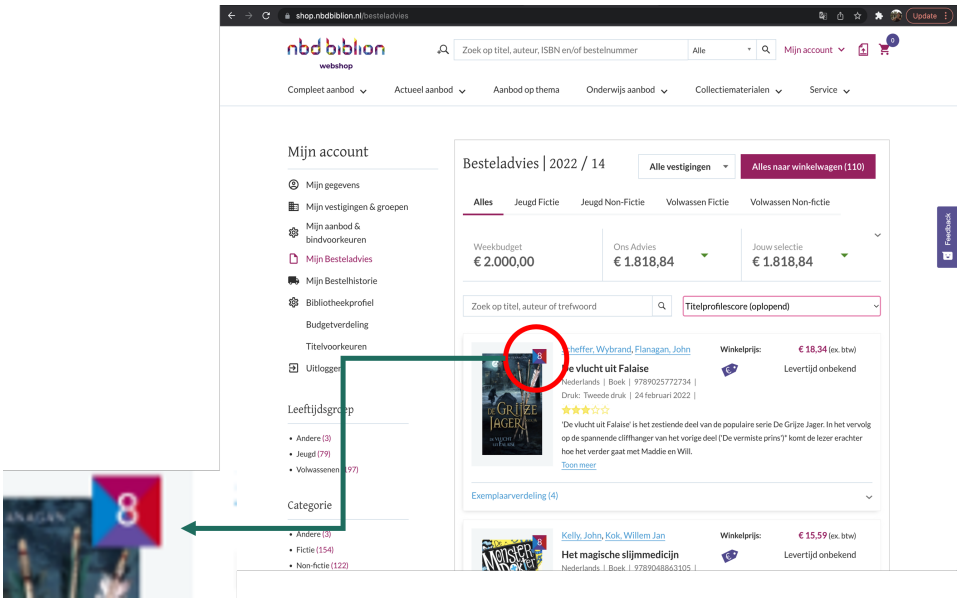

Figure 7: Current implementation of the generic popularity score within the platform of NBD Biblion.

B *Construction of the target label*

B.1 *Transaction records*

Table 9: Transaction records for one book across three libraries. This includes the identifier (PPN), the number of books, the library and which type of transaction it is.

PPN	Amount	Library	Type
865881111	1	NL-0800790000	Reservation
865881111	2	NL-0800790000	Loan
865881111	1	NL-0800790000	Reservation
865881111	2	NL-0830860000	Loan
865881111	2	NL-0830860000	Reservation
865881111	1	NL-0830860000	Loan
865881111	1	NL-0873440000	Loan
865881111	1	NL-0873440000	Reservation
865881111	2	NL-0873440000	Loan

B.2 *Inventory records*

Table 10: Inventory records for one book across three libraries, measured at multiple dates. This includes the identifier (PPN), the number of books, the library and the measurement date.

PPN	Amount	Library	Date
865881111	1	NL-0800790000	7-1-2021
865881111	2	NL-0800790000	8-1-2021
865881111	1	NL-0800790000	10-1-2021
865881111	1	NL-0830860000	7-1-2021
865881111	3	NL-0830860000	8-1-2021
865881111	3	NL-0830860000	10-1-2021
865881111	4	NL-0873440000	7-1-2021
865881111	4	NL-0873440000	8-1-2021
865881111	3	NL-0873440000	10-1-2021

B.3 *Encoded transaction table*

Table 11: Transaction table encoded according to *type* variable, which multiplies *loan* times 1.5.

PPN	Amount	Library	Type_loan	Type_reservation
865881111	1	NL-0800790000	1	
865881111	2	NL-0800790000		3
865881111	1	NL-0800790000	1	
865881111	2	NL-0830860000		3
865881111	2	NL-0830860000	2	
865881111	1	NL-0830860000		1.5
865881111	1	NL-0873440000		1.5
865881111	1	NL-0873440000	1	
865881111	2	NL-0873440000		3

B.4 *Transaction and inventory ratio*

Table 12: Transaction and inventory ratio, calculated by dividing the summed *Reservation Loan* variables by the average inventory per library.

PPN	Loan Reservation	Inventory (avg)	Trans_invt ratio	Library:
865881111	5	4	1.25	NL-0800790000
865881111	6.5	7	0.92	NL-0830860000
865881111	5.5	11	0.5	NL-0873440000

B.5 *Inter ratio*

Table 13: The *Inter ratio* variable is calculated by dividing the sum of the book-specific *Trans_invt* ratio by the *Trans_invt ratio*.

PPN	Sum_trans_invt ratio	Trans_invt ratio	Library	Inter_ratio
865881111	2.67	1.25	NL-0800790000	0.47
865881111	2.67	0.92	NL-0830860000	0.35
865881111	2.67	0.5	NL-0873440000	0.19

B.6 *Robust Scaler*

Table 14: Robust Scaler applied on the *Inter_ratio* variable.

PPN	Inter_ratio	Library	Robust scaler
865881111	0.47	NL-0800790000	0.86
865881111	0.35	NL-0830860000	0
865881111	0.19	NL-0873440000	-1.14

C *Influence of data-driven score*

Table 15: Class labels and their influence on the generic popularity score.

Class	**Add or Subtract**
0	- 1
1	- 0.5
2	0.5
3	1

D *Distribution of the training set with and without SMOTE*

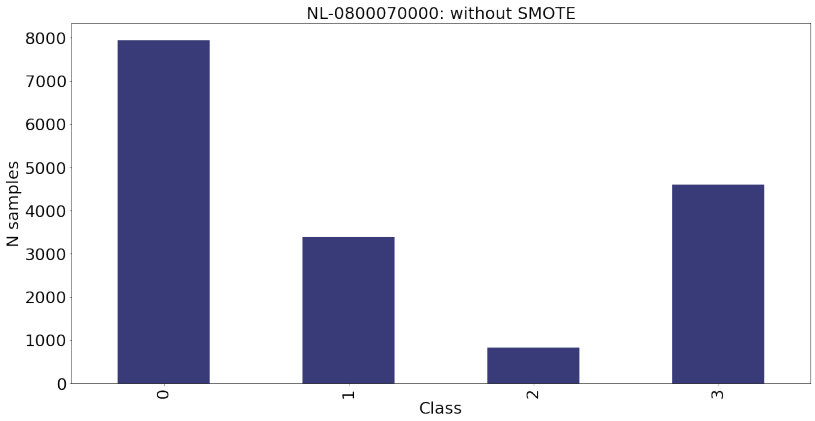

Figure 8: Distribution of the target label within library NL-0800070000, before SMOTE is applied.

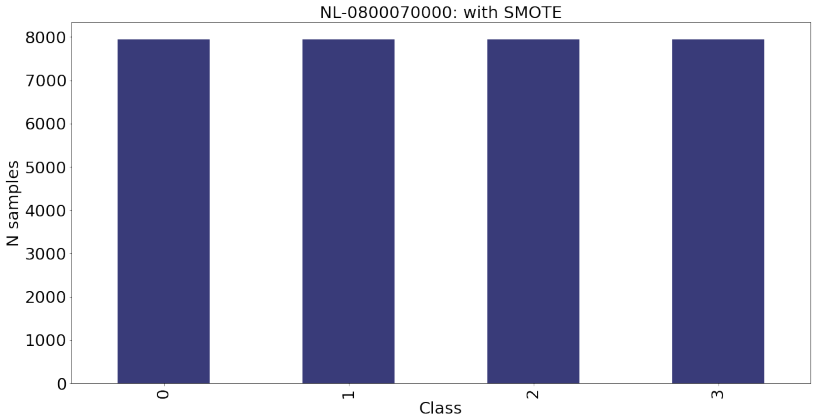

Figure 9: Distribution of the target label within library NL-0800070000, when SMOTE is applied.

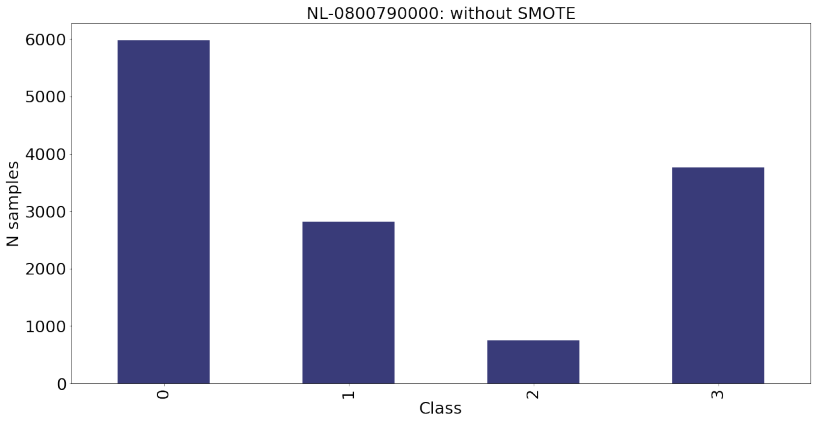

Figure 10: Distribution of the target label within library NL-0800790000, before SMOTE is applied.

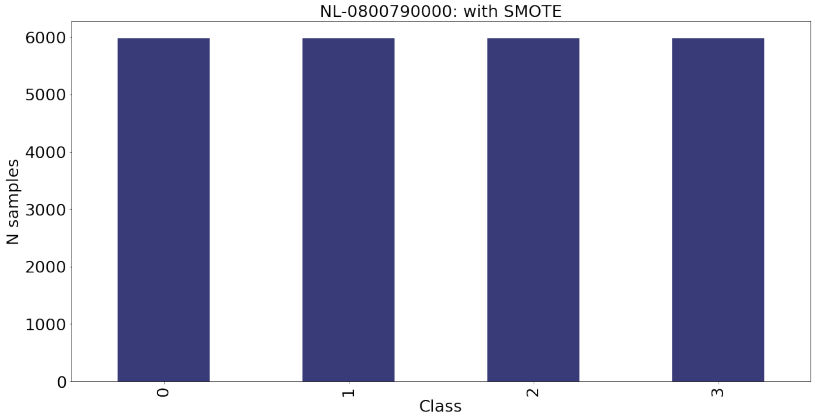

Figure 11: Distribution of the target label within library NL-0800790000, when SMOTE is applied.

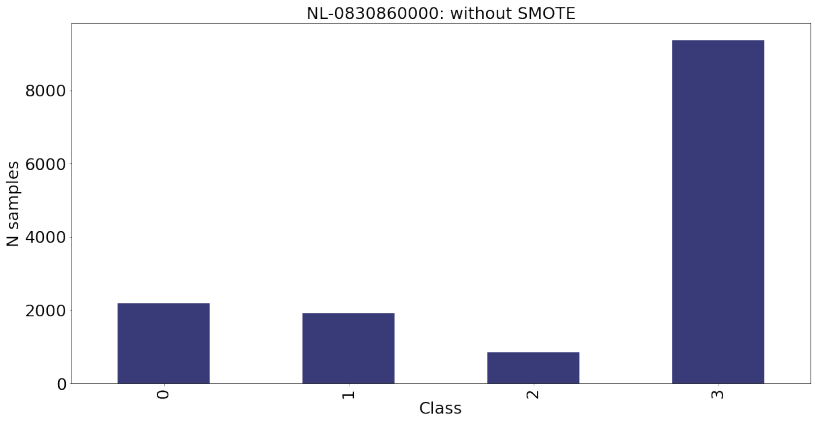

Figure 12: Distribution of the target label within library NL-0830860000, before SMOTE is applied.

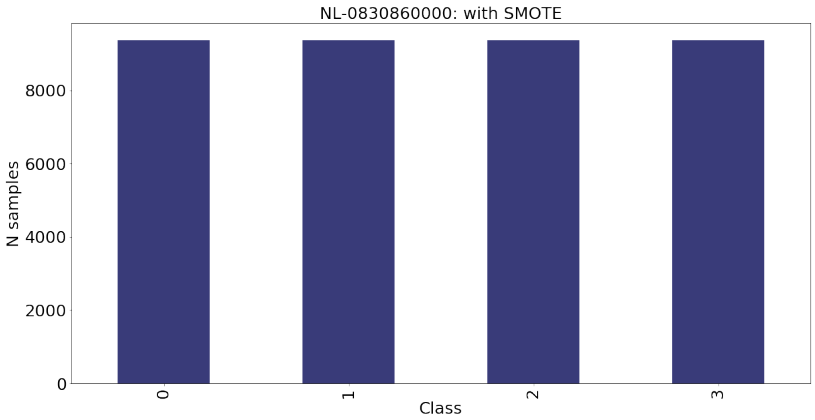

Figure 13: Distribution of the target label within library NL-0830860000, when SMOTE is applied.

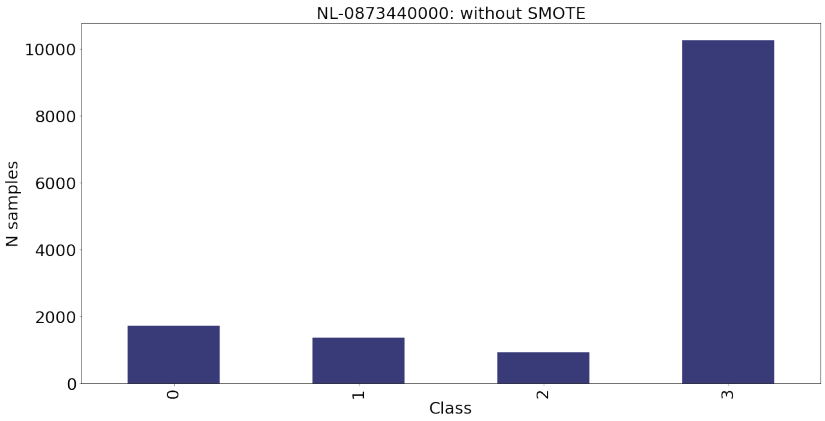

Figure 14: Distribution of the target label within library NL-0873440000, before SMOTE is applied.

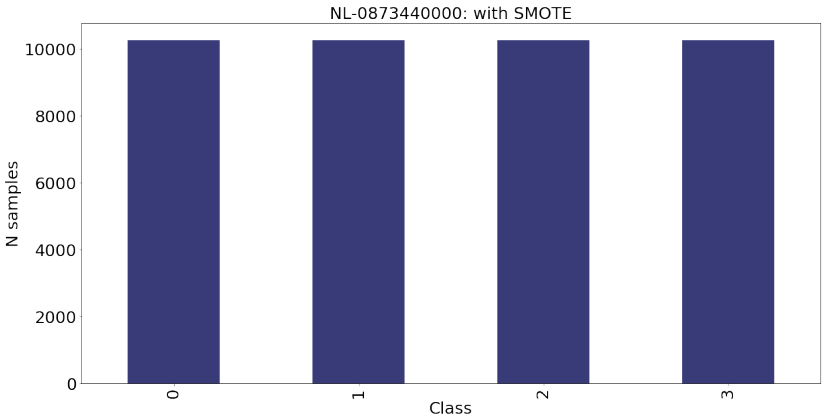

Figure 15: Distribution of the target label within library NL-0873440000, when SMOTE is applied.

E *Distribution of the target label before and after applying SMOTE*

Table 16: The number of data points before and after applying SMOTE on the training set, including the total number of instances that were added by SMOTE. This is described per class (which are: 0, 1, 2, 3).

Library	**Before SMOTE**			
	0	1	2	3
NL-0800070000	7941	3388	829	4606
NL-0800790000	5978	2812	751	3759
NL-0830860000	2192	1912	853	9360
NL-0851030000	3102	1921	1042	12166
NL-0873440000	1730	1370	931	10261
	After SMOTE			
	0	1	2	3
NL-0800070000	7941	7941	7941	7941
NL-0800790000	5978	5978	5978	5978
NL-0830860000	9360	9360	9360	9360
NL-0851030000	12166	12166	12166	12166
NL-0873440000	10261	10261	10261	10261
	Added by SMOTE			
	0	1	2	3
NL-0800070000	0	4553	7112	3335
NL-0800790000	0	3166	5227	2219
NL-0830860000	7168	7448	8507	0
NL-0851030000	9064	10245	11124	0
NL-0873440000	8531	8891	9330	0

F *Grid Search output per library and model*

Table 17: Results Grid Search per model for library: NL-800070000.

NL-800070000					
XGBOOST		**SVM**		**LR**	
Hyper- parameter	Value	hyperparameter	Value	Hyper- parameter	Value
Max depth	25	C	10	Solver	newton-cg
Eta	0.1	Gamma	0.1	Penalty	l2
N estimators	100	Kernel	rbf	C	1
Colsample by tree	0.8			Max iter	500
Gamma	0.5				
Subsample	0.8				

Table 18: Results Grid Search per model for library: NL-800079000.

NL-800079000					
XGBOOST		**SVM**		**LR**	
hyperparameter	Value	hyperparameter	Value	hyperparameter	Value
Max depth	15	C	10	Solver	lbfgs
Eta	0.1	Gamma	0.1	Penalty	None
N estimators	300	Kernel	rbf	C	0.01
Colsample by tree	0.8			Max iter	500
Gamma	0.5				
Subsample	0.6				

Table 19: Results Grid Search per model for library: NL-083086000.

NL-083086000					
XGBOOST		**SVM**		**LR**	
hyperparameter	Value	hyperparameter	Value	hyperparameter	Value
Max depth	25	C	1	Solver	saga
Eta	0.1	Gamma	0.1	Penalty	l2
N estimators	100	Kernel	rbf	C	0.01
Colsample by tree	0.8			Max iter	500
Gamma	0.5				
Subsample	0.8				

Table 20: Results Grid Search per model for library: NL-0851030000.

NL-0851030000					
XGBOOST		**SVM**		**LR**	
hyperparameter	Value	hyperparameter	Value	hyperparameter	Value
Max depth	25	C	10	Solver	saga
Eta	0.1	Gamma	0.1	Penalty	None
N estimators	100	Kernel	rbf	C	0.01
Colsample by tree	0.8			Max iter	500
Gamma	0.5				
Subsample	0.8				

Table 21: Results Grid Search per model for library: NL-0873440000.

NL-0873440000					
XGBOOST		**SVM**		**LR**	
hyperparameter	Value	hyperparameter	Value	hyperparameter	Value
Max depth	25	C	10	Solver	newton-cg
Eta	0.1	Gamma	0.01	Penalty	None
N estimators	300	Kernel	rbf	C	0.01
Colsample by tree	0.6			Max iter	1000
Gamma	0.5				
Subsample	0.8				

G *Macro F1 scores of the Baseline-experiment and SMOTE-experiment*

Table 22: Macro F1 scores per library and model concerning the Baseline-experiment. Here the highest values per library are marked bold.

	Macro F1 scores of the Baseline-experiment		
Library	XGB	SVM	LR
NL-0800070000	**0.35**	0.29	0.22
NL-0800790000	**0.31**	0.29	0.30
NL-0830860000	**0.33**	0.30	0.32
NL-0851030000	0.29	**0.33**	0.27
NL-0873440000	**0.33**	**0.33**	0.18

Table 23: Macro F1 scores per library and model concerning the SMOTE-experiment. Here the highest values per library are marked bold.

	Macro F1 scores of the SMOTE-experiment		
Library	**XGB**	**SVM**	**LR**
NL-0800070000	**0.33**	0.27	0.25
NL-0800790000	**0.30**	0.27	0.28
NL-0830860000	**0.32**	**0.32**	0.31
NL-0851030000	**0.31**	0.27	0.29
NL-0873440000	**0.29**	0.22	0.24

H *Feature importances of the XGB-baseline models*

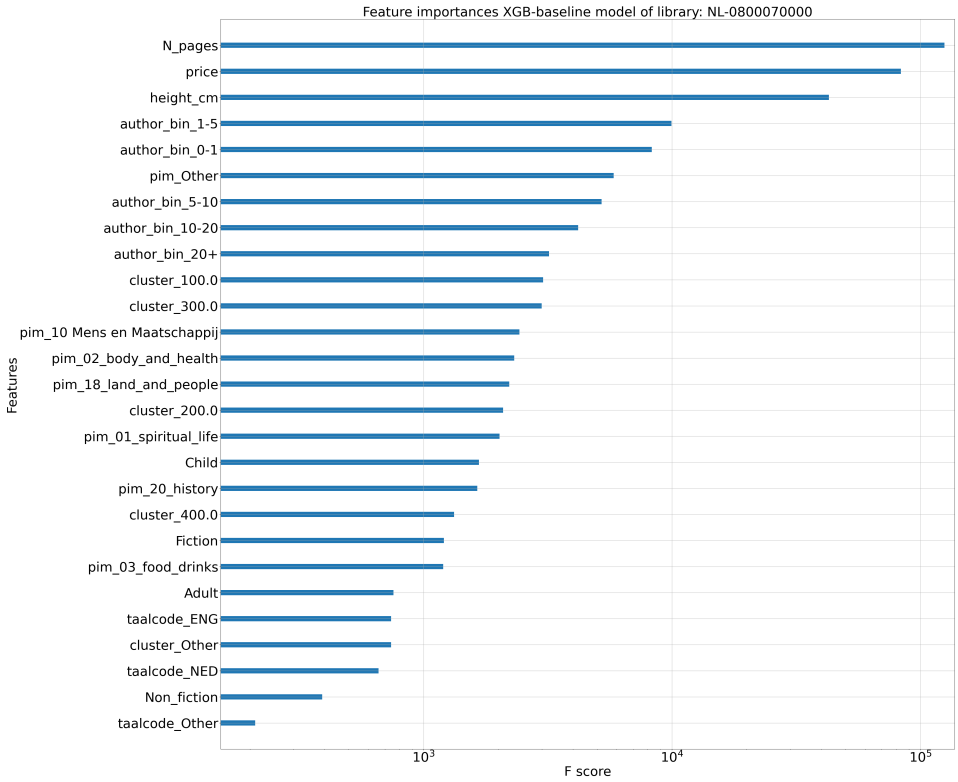

Figure 16: XGB-baseline Feature importances of library: NL-0800070000.

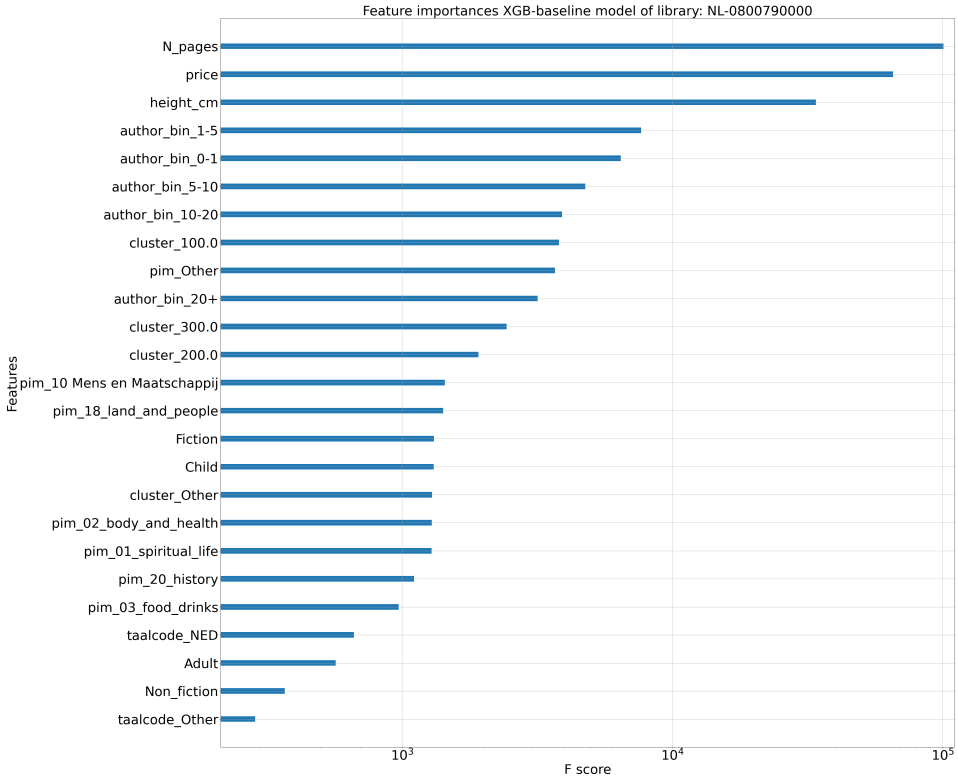

Figure 17: XGB-baseline Feature importances of library: NL-080079000.

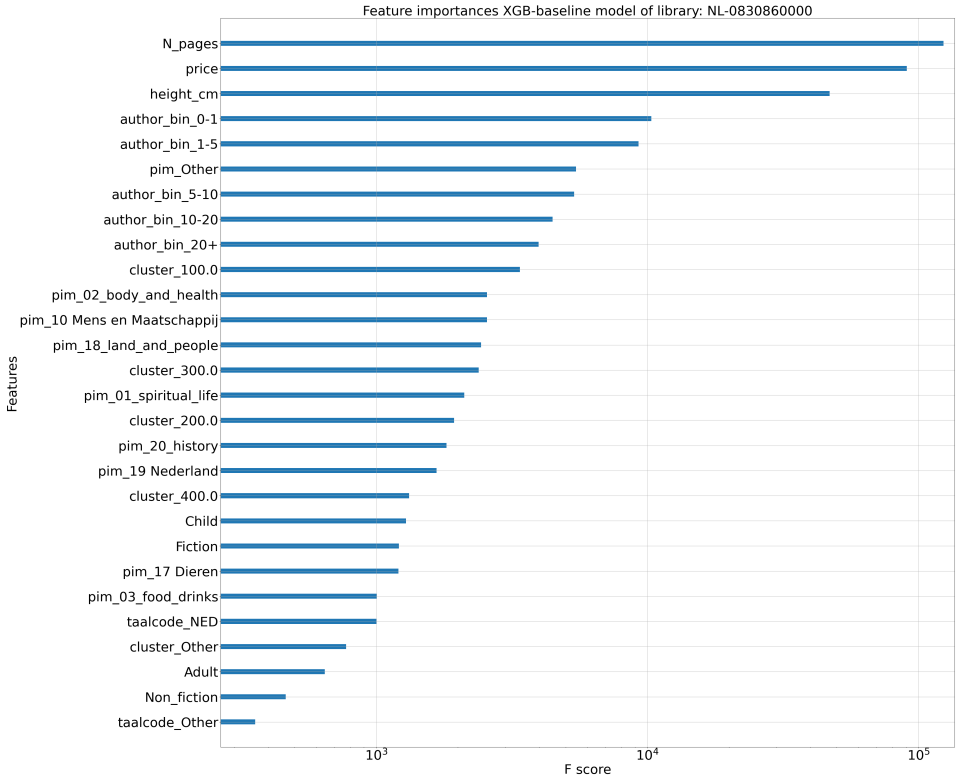

Figure 18: XGB-baseline Feature importances of library: NL-0830860000.

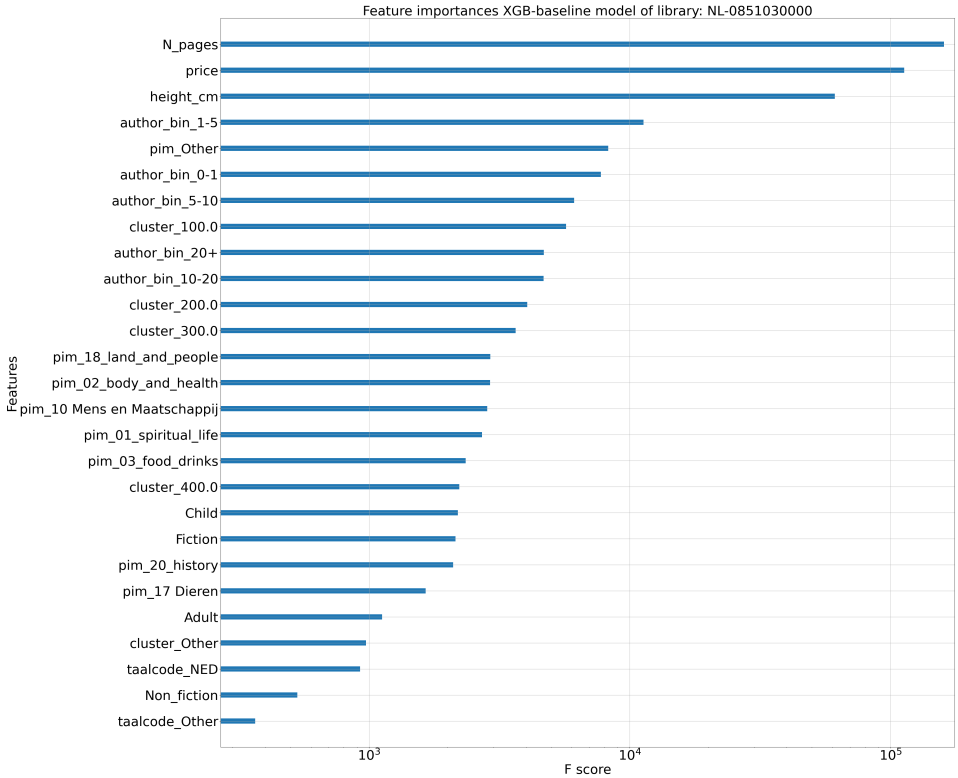

Figure 19: XGB-baseline Feature importances of library: NL-0851030000.

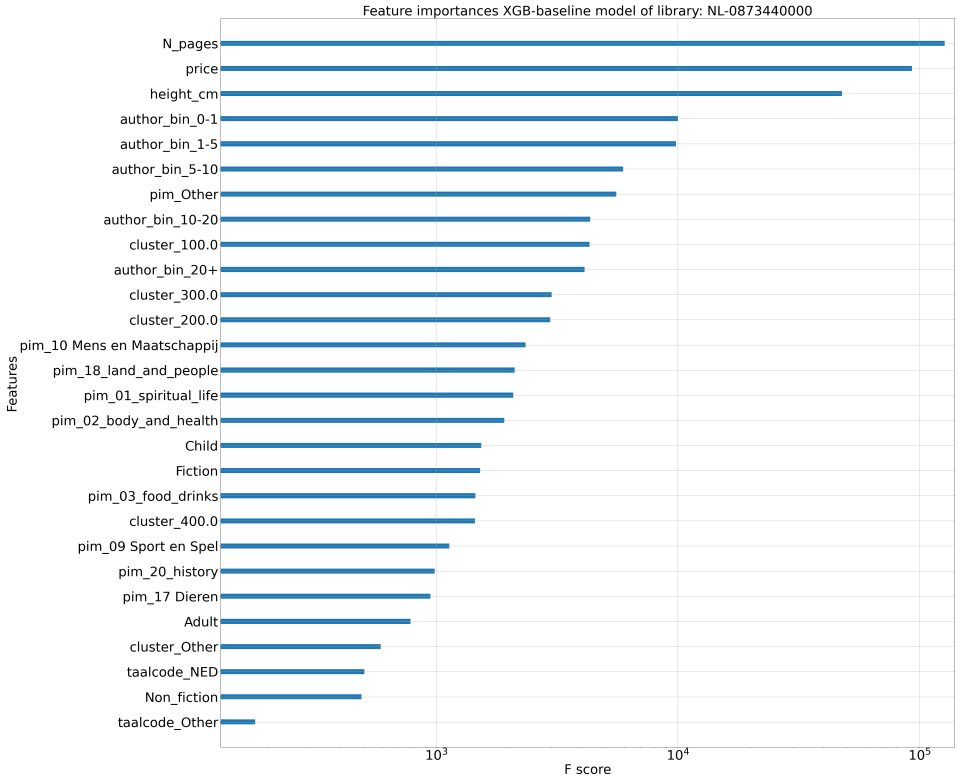

Figure 20: XGB-baseline Feature importances of library: NL-0873440000.